

Scope Planning

Project success is determined by its usefulness or profitability:

- in increase of revenue
- in savings of costs

The main reason to change existent information system is to get more benefits to organization, to help more to achieve its strategic goals obtainable benefits must be expressed in information system (new, changed) goals. After the project completion developed information system must meet these requirements what implement information system goals.

Here raises a question: “what are these requirements to what developed information system must meet?” More precisely:

- What are the projects deliverables?
 - What kind of must be the constitution of changed (future) information system to achieve goals expressing information system value?
- What customer really wants?
- What are formalities to take into account?

In the context of project management with expected outcome and its requirements deals scope management. In the context of information system we can take scope as all requirements what developed information system can meet. This concept is illustrated on the next figure:

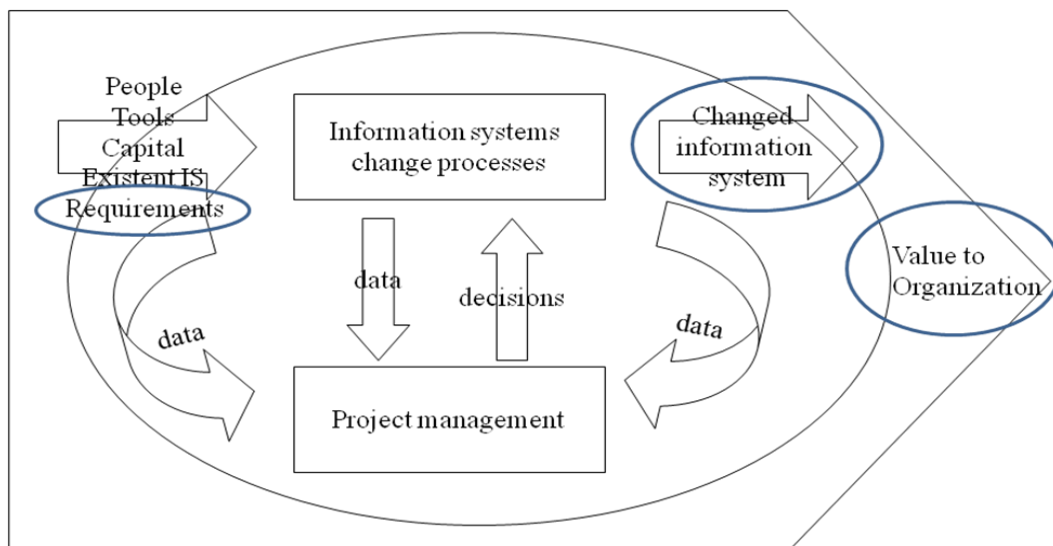


Figure 1. Scope Context in the Information System Development

Scope Definition

Scope is the sum of the products, services, and results to be provided as a project (PMBOK). In the context of project management we must define, write down and get products owner agreement for 2 things:

- to “breadth” of expected deliverables – product (hereby system) scope
- to “depth” of expected deliverables – project scope

Product scope - the features and functions that characterize a product, service, or result

Project scope - the work that needs to be accomplished to deliver a product, service, or result with the specified features and functions

Product Scope

In defining scope of project deliverables we must agree with project stakeholders (customer and performer) and write down unambiguously following aspects:

- what product, service, or result will do
- how the product, service, or result will be used
- how the product, service, or result will function
- what the product will look like, what the service is, or what the result will be
- what impact the product, service, or result will have on the organization, customer, stakeholders, and business processes
- any constraints, restrictions, standards, regulations, and other requirements related to the product

Scope of changeable information system– amount of aspects or range of IS architecture (what will be in and out of borders) affected with change including:

- quantity of IS goals
- quantity of IS processes
- quantity of actors in IS
- quantity of functional/non-functional requirements
- quantity of data entities
- quantity of locations

Project Scope

Project scope deals with how one or another expected deliverable will be created and delivered to customer. It deals with deliverables and project management documentation describing all the work we must accomplish to achieve project goals. Depending on customer technical experience, documentation can be more or less complex or formal, for example:

- data model where are shown main data entities with relations or logical data model where data are normalized with 3. normal form
- application specification which consists of manually written sentences on paper or of to the very last elaborated diagram showing program structure

Different choices demand different amount of work

Additionally – in context of system development, project scope is dictated from system development methodology determining, what kind of documentation is needed, in which sequence and content. With project scope we must determine, what kind of methodology customer uses and how strictly it will be followed. If client considers that following methodology is costly, even so it is necessary to follow it in appropriate level when to consider project success.

One example of quantity of rules to be followed by certain methodology showing on the next figure:

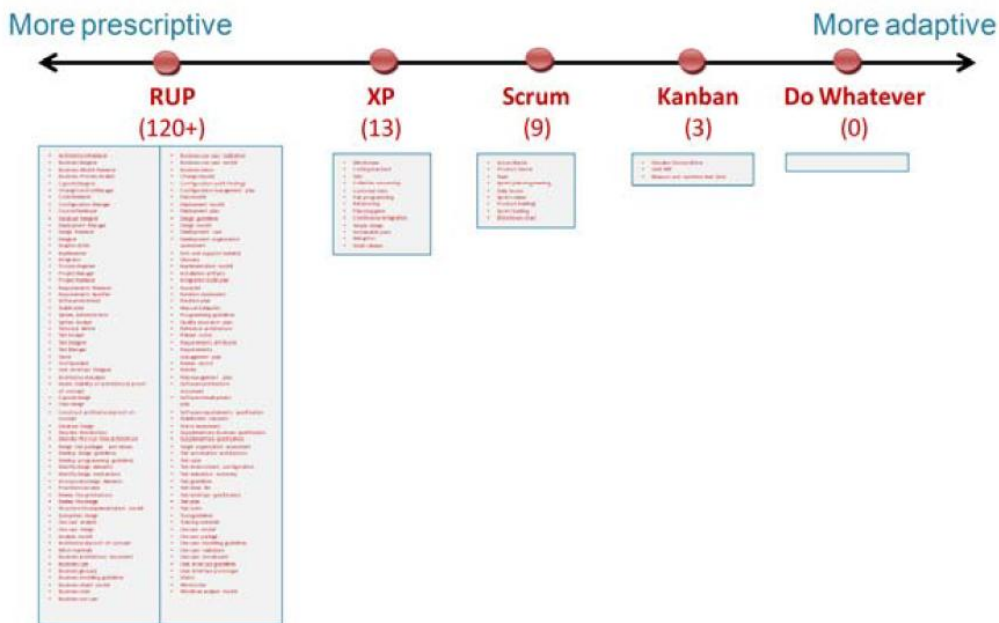


Figure 2. Methodology Rules Quantity Difference

In context of system development project scope by RUP includes:

- customer products (things what customer needs for useful system, in addition to program code also user manuals, help files, installation scripts and guides)
- process artifacts (things what development team during development creates, for example use case specifications, design documentation etc)
- internal deliverables (things having value to development organization – development results consisting intellectual value, for example requirements documentation templates, test plans etc)

- services (work what is expected in addition to delivered product or development of another results, for example training, consultation, installation, onsite support and adaptation to customer)

Scope Management

The main question to project is: “are we doing the RIGHT thing?”. Nobody wants to find itself in situation where client says: “but I thought, I was getting “X!””. With answering to this question deals scope management including scope planning and controlling scope against (quantitative) changes.

In PMBOK it is named “project scope management” and it includes processes required to ensure that **the project includes all the work required, and only the work required, to complete the project successfully**. Managing the project scope is primarily concerned with defining and controlling **what is and is not** included in the project

Scope Management Processes in PMBOK

Scope management processes are presented on the next figure. The circle is drawn over scop planning processes what are the subject of this lecture.

	Initiating	Planning	Executing	Monitoring & Controlling	Closing
Scope		Collect Requirements Define Scope Create WBS		Verify Scope Control Scope	

Figure 3. Scope Management Processes

These processes are discribed in scope management plan that provides guidance on how project scope will be defined, documented, verified, managed, and controlled. Management object is baselined scope consisting of the approved project scope statement and its associated WBS and WBS dictionary. This baselined scope is then monitored, verified, and controlled throughout the lifecycle of the project

Scope Planning

Scope planning processes, inputs and outputs are presented on the next figure:

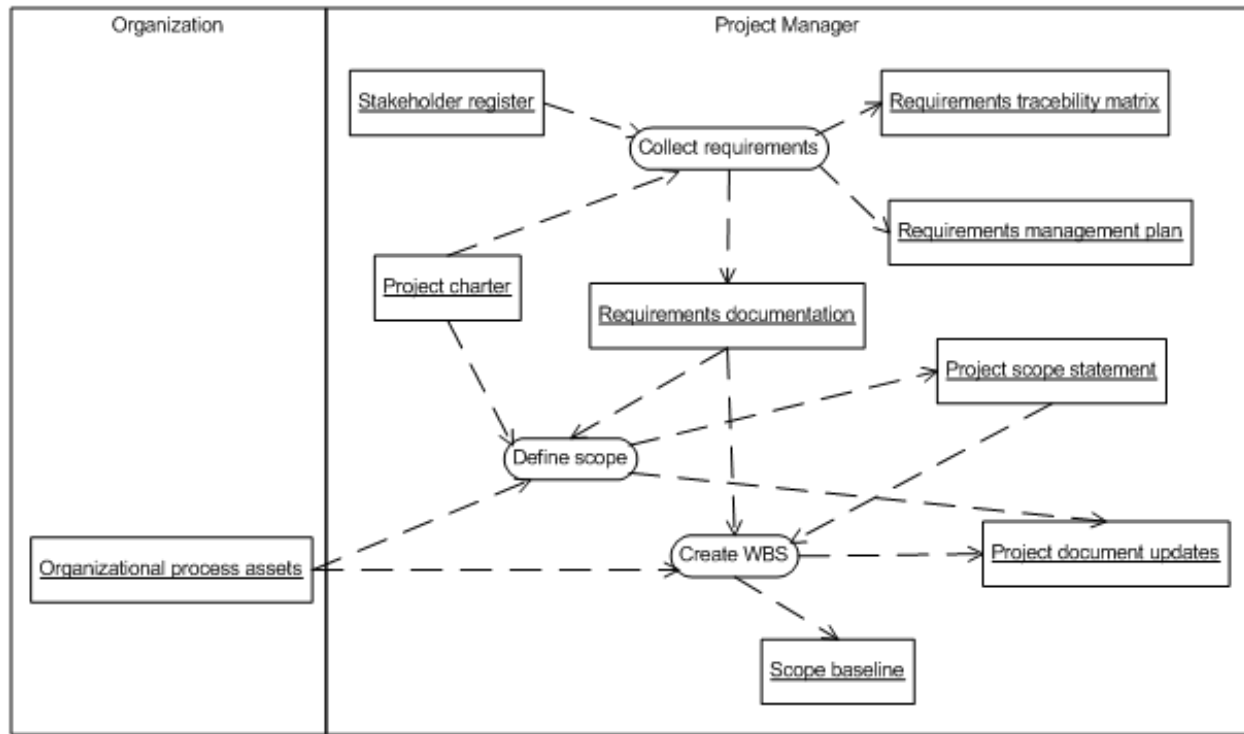


Figure 4. Scope Planning Processes, Inputs and Outputs

Scope planning processes are

- collecting requirements
- define scope
- creating work breakdown structure – WBS

Collect Requirements

Collecting requirements is the process of defining and documenting stakeholders' needs to meet the project objectives. The project's success is directly influenced by the care taken in capturing and managing project and product requirements. Requirements include the quantified and documented needs and expectations of the sponsor, customer, and other stakeholders. Collecting requirements is defining and managing customer expectations. Requirements become the foundation of the WBS, cost, schedule, and quality planning.

Inputs for this process are project charter and stakeholder register and outputs are requirements documentation; requirements management plan and requirements

traceability matrix. Tools and techniques for collecting requirements are system analysis methods and techniques. Remark: system development methodology used in the project must give guidelines for requirements collecting, documenting and tracing their life cycle.

Requirements Documentation

Requirements documentation describes how individual requirements meet the business need for the project. Requirements may start out at a high level and become progressively more detailed as more is known. Before being baselined, requirements must be unambiguous (measurable and testable), traceable, complete, consistent, and acceptable to key stakeholders – SMART. The format of a requirements document may range from a simple document listing all the requirements categorized by stakeholder and priority, to more elaborate forms containing executive summary, detailed descriptions, and attachments

Components of Requirements Documentation may be as follows:

- Business need or opportunity to be seized, describing the limitations of the current situation and why the project has been undertaken;
- Business and project objectives for traceability;
- Functional requirements, describing business processes, information, and interaction with the product, as appropriate which can be documented textually in a requirements list, in models, or both;
- Non-functional requirements, such as level of service, performance, safety, security, compliance, supportability, retention/purge, etc.;
- Quality requirements;
- Acceptance criteria;
- Business rules stating the guiding principles of the organization;
- Impacts to other organizational areas, such as the call center, sales force, technology groups;
- Impacts to other entities inside or outside the performing organization;
- Support and training requirements;
- Requirements assumptions and constraints

Requirements Management Plan

Documents how requirements will be analyzed, documented, and managed throughout the project. Components of the requirements management plan can include, but are not limited to:

- How requirements activities will be planned, tracked, and reported;
- Configuration management activities;

- Requirements prioritization process;
- Product metrics that will be used and the rationale for using them;
- Traceability structure - which requirements attributes will be captured on the traceability matrix and to which other project documents requirements will be traced

Requirements Traceability Matrix

A table that links requirements to their origin and traces them throughout the project life cycle. The implementation of that matrix helps ensure that each requirement adds business value by linking it to the business and project objectives. It provides a means to track requirements throughout the project life cycle, helping to ensure that requirements approved in the requirements documentation are delivered at the end of the project. It provides a structure for managing changes to the product scope

This process includes, but is not limited **to tracing**:

- Requirements to business needs, opportunities, goals, and objectives;
- Requirements to project objectives;
- Requirements to project scope/WBS deliverables;
- Requirements to product design;
- Requirements to product development;
- Requirements to test strategy and test scenarios; and
- High-level requirements to more detailed requirements

Requirements Attributes

Attributes associated with each requirement can be recorded in the requirements traceability matrix. These attributes help to define key information about the requirement. Typical attributes used in the requirements traceability matrix may include: a unique identifier, a textual description of the requirement, the rationale for inclusion, owner, source, priority, version, current status (such as active, cancelled, deferred, added, approved) and date completed. Additional attributes to ensure that the requirement has met stakeholders' satisfaction may include stability, complexity, and acceptance criteria.

Risks associated with deliverable uncertainty

At first sight seems that definition of deliverables is simple. Customer wants amount of models, technological architecture or application system, which consists of code and documentation. Creating these things may be difficult, defining these things seems easy. Unfortunately in projects world simplicity leads to abyss. Problems rise from

seemingly safe requirements. For example: “new warehouse system will simplify financial data processing for accounting system”. This can mean, that ...

- “new system will output reports where are shown summary data, what must be enter manually to account system” or
- “new system will in the end of every month generate data file, which will transported to account system” or
- “account systems database will be updated from warehouse system online”

Work what must be carried out corresponding these interpretations is different. When project manager plans as solution reports in the month end, but customer wants online updating, then trouble is in house.

Conclusion: before beginning development work we must understand not only what all these requirements mean, but also, what customer thinks, what these requirements mean

Defining Scope

Defining scope is creating project scope statement. It is a process of developing a detailed description of the project and product. Inputs for tgis process are project charter, requirements documentation and organization process assets (templates for example). The output of this process is project scope statement. The preparation of a detailed project scope statement is critical to project success and builds upon the major deliverables, assumptions, and constraints that are documented during project initiation. During planning, the project scope is defined and described with greater specificity as more information about the project is known. Existing risks, assumptions, and constraints are analyzed for completeness; additional risks, assumptions, and constraints are added as necessary. If something isn't described within the detailed project scope statement then the work should not be done or the scope statement needs revised to include the work

Project Scope Statement

Describes, in detail, the project's deliverables and the work required to create those deliverables; **provides** a common understanding of the project scope among project stakeholders; **may contain** explicit scope exclusions that can assist in managing stakeholder expectations; **enables** the project team to perform more detailed planning, **guides** the project team's work during execution, **provides** the baseline for evaluating whether requests for changes or additional work are contained within or outside the project's boundaries.

The degree and level of detail to which the project scope statement defines the work that will be performed and the work that is excluded can determine how well the project management team can control the overall project scope

Project scope statement should consist at least following components (they may be written in separate documents):

- Product scope: The characteristics of the product, service, or result for which the project was undertaken. In projects that are part of a larger program, the project itself may only be creating components of the product, but the product scope or product description is still necessary so that everyone knows what the overall objective is.
- Project objectives: Objectives are the success metric for the project. Specifically, what will it take for the project to be considered successful? This includes the business, cost, schedule, technical, and quality objectives, and other specific targets should be included where applicable.
- Project requirement: The capabilities that the product, service, or result must possess and meet. Requirements are the translated expectations and needs of the stakeholders into prioritized, descriptive requirements and work items.
- Project exclusions: Nearly just as important as what IS in the project, the scope should include items that are excluded from the project. Doing this helps eliminate any confusion within the stakeholders or project team.
- Project deliverables: The core product, service, or result should be fully described, as well as any ancillary deliverables. Any needed project artifacts, those documents not directly related to the deliverables, such as management, technical, or status reports, should also be described.
- Product acceptance criteria: The process and criteria for product acceptance should be defined. This includes customer-specific requirements and any testing or other threshold limits.
- Project constraints: Any limiting factors the project must work within, such as deadlines, budget, staffing, facilities, equipment, materials, or contractual restraints, should be described.
- Project assumptions: Every project has assumptions, and these should be described because assumptions are risk factors.
- Risks: Risks should be identified at least at a high level. The risk register is where all risks are logged, but having major risks explained in the scope statement helps make everyone aware and on the lookout for them.
- Milestones: Any important dates, including deliverable- or artifact-oriented dates should be included in the project scope statement.

- Approval requirements: Any specific approval requirements for items such as deliverables, documents, and work should be described.

Creating Work Breakdown Structure

Deliverables-oriented, graphical, hierarchical representation of the work required to fulfill the project scope statement. Purposes:

- it subdivides the work into manageable components that can be scheduled, estimated, and assigned
- through the process of creating and updating the WBS, it helps to identify needed work that might otherwise not have not been discovered until later
- it can be used as a visual communication tool for the customer, stakeholders, and project team
- the WBS is an input to activity definition, cost estimating, cost budgeting, resource planning, and risk management planning

Inputs for creating WBS are project scope statement; requirements documentation and organizational process assets (templates). Outputs are WBS, WBS dictionary, scope baseline and project documents updates. Tool and technique for that process is decomposition.

WBS Design Principles

- Project scope is divided into manageable components in terms of size, duration, and responsibility (e.g., systems, subsystems, components, tasks, subtasks, and work packages) which include all steps necessary to achieve the objective – result is hierarchy or tree
- WBS includes 100% of the work defined by the project scope and captures ALL deliverables – internal, external, interim – in terms of the work to be completed, including project management
- The sum of the work at the “child” level must equal 100% of the work represented by the “parent”
- WBS should not include any work that falls outside the actual scope of the project, that is, it cannot include more than 100% of the work...
- It is important that there is no overlap in scope definition between two elements of a Work Breakdown Structure

Examples of creating WBS structure:

- Using phases of the project life cycle as the first level of decomposition, with the product and project deliverables inserted at the second level;
- Using major deliverables as the first level of decomposition;
- Using subprojects which may be developed by organizations outside the project team, such as contracted work - the seller then develops the supporting contract work breakdown structure as part of the contracted work

Example of WBS pictorially:

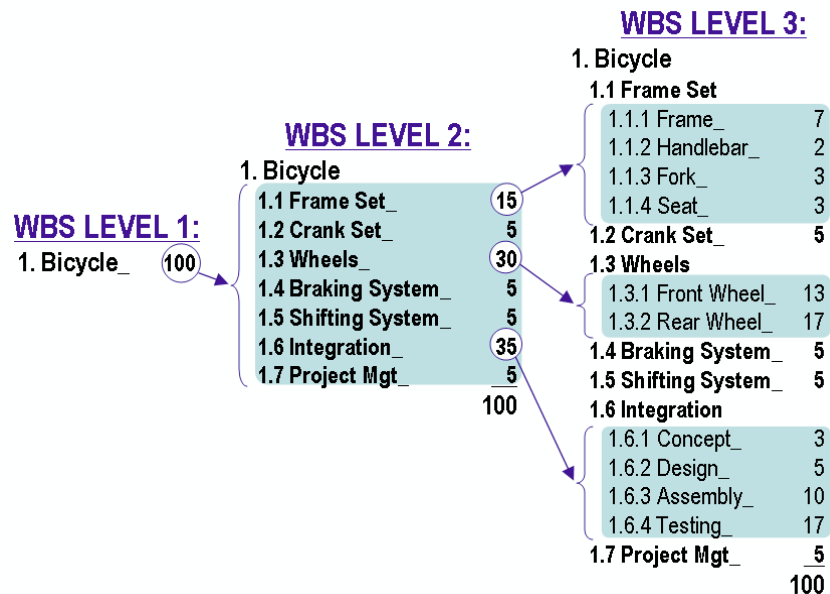


Figure 5. Example of Bicycle Work Breakdown Structure

WBS Dictionary

Provides more detailed descriptions of the components in the WBS, including work packages and control accounts:

- Code of account identifier
- Description of work
- Responsible organization
- List of schedule milestones
- Associated schedule activities
- Resources required
- Cost estimates
- Quality requirements
- Acceptance criteria
- Technical references

- Contract information

Scope Baseline

A component of the project management plan includes project scope statement; WBS and WBS dictionary.

Scope Planning in RUP

With project scope defining deals environment discipline and in that workflow named “develop development case” determining what artifacts must be developed and in what accuracy. With defining and management of requirements deals requirements discipline and in that more specifically “define the system”, “manage the system scope” and “refine the system definition” workflows.

Scope Planning in Agile Approach by Example of SCRUM

Consists of following activities:

- Perform domain analysis for building domain model
- Development of a comprehensive product backlog list
- Development of a comprehensive product sprint backlog
- Definition of the functionality that will be included in each release
- Selection of the release most appropriate for immediate development

Criteria for Choosing System Development Methodology

- Nature and scope of system under development
- Project criticality
- Budget
- Team Size
- Used Technology
- Used Tools and Techniques
- Work culture in organization

Nature of the Software Developments

Steve McConnell in his book "Rapid Development" differentiates basic lifecycle approaches to 3 groups:

- Demos and “proof-of-concept” prototyping - needs some conceptual modeling to guide the coding, but not detailed requirements, design, or testing
- New technology application delivery - requires an incremental approach, iterating between the software development phases of architecture, detailed design, coding, and testing

- Stable application maintenance - needs a sequential waterfall approach repeating a well-understood process for a well-understood product or application

Adaptive (“agile”) versus Predictive (“plan-driven”) Approach

- Adaptive/ agile
 - Low criticality
 - Senior developers
 - Requirements change often
 - Small number of developers
 - Culture that thrives on chaos
- Predictive/ plan-driven
 - High criticality
 - Junior developers
 - Requirements do not change often
 - Large number of developers
 - Culture that demands order

Methodologies, organized as people x criticality x optimization

Project criticality and corresponding team size with project management formality is presented on the next figure:

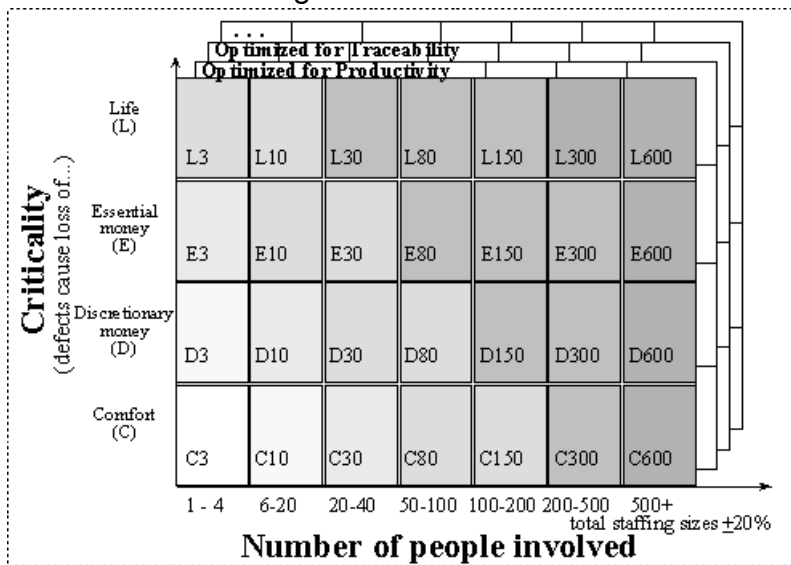


Figure 6. Project Criticality related with Team Size

Boehm and Turner Radar Chart

- Personnel - measures team skills
- Dynamism - likelihood of changes
- Culture - temperament of the organization – thriving on chaos vs. order
- Team Size
- Criticality - system failure results in loss of ...

Pictorially expressing:

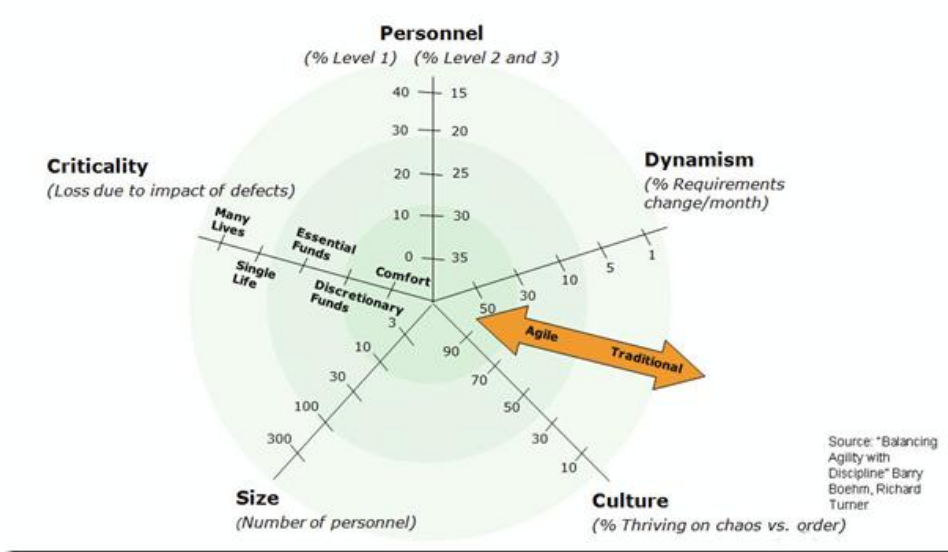


Figure 7. Boehm and Turner Radar Chart

Instead of Summary

Henrik Kniberg and Mattias Skarin in their book "From Kanban and Scrum. Making the Most of Both" are written following:

- All system methodologies are process tools
 - Tools – anything used as a means of accomplishing a task or purpose
 - Process - how you work
- They help you work more effectively by, to a certain extent, telling you what to do
- Using the right tools will help you succeed, but will not guarantee success
- It's easy to confuse project success/failure with tool success/failure
 - A project may succeed because of a great tool
 - A project may succeed despite a lousy tool
 - A project may fail because of a lousy tool
 - A project may fail despite a great tool
- We can compare process tools by looking at how many rules they provide where prescriptive means "more rules to follow" and adaptive means "fewer rules to follow"
 - 100% prescriptive means you don't get to use your brain, there is a rule for everything
 - 100% adaptive means Do Whatever, there are no rules or constraints
- No tool is complete, no tool is perfect
- Compare for understanding, not for judgment
- Don't limit yourself to one tool!
- Mix and match the tools as you need!

Used Literature

- Project Management Institute, A Guide to the Project Management Body of Knowledge (PMBOK® Guide) - Fourth Edition
- Scope Planning Templates, <http://www.projectmanagementdocs.com/project-planning.html>
- Jolyon E. Hallows: Information Systems Project Management, 1998
- Simon Harris, The Breakdown Structure: 90% of how to manage projects safely, <http://www.asapm.org/asapmag/articles/TheBreakdownStructurePt1.pdf> ja <http://www.asapm.org/asapmag/articles/TheBreakdownStructurePt2.pdf>
- WBS näiteid
 - http://en.wikipedia.org/wiki/Work_breakdown_structure
 - <http://www.pmhut.com/project-management-process-phase-2-planning-develop-project-schedule>
- WBS Review, http://cvisn.fmcsa.dot.gov/downdocs/cvisndocs/plan99/2%20Day%2000%20Sessions%20for%20update/P2%20Intro%20to%20WBS%20Development_R26.pptRUP, <http://sce.uhcl.edu/helm/rationalunifiedprocess/index.htm>
- P. Fitsilis, Comparing PMBOK and Agile Project Management Software Development Processes, <http://www.springerlink.com/content/j52npr0157v26386/>
- William J. Brown, Hays W. McCormick III, Scott W. Thomas: Antipatterns in Project Management, 2000
- Alistair Cockburn, Methodology per project, <http://alistair.cockburn.us/Methodology+per+project>
- Mike Griffiths, “Agile Suitability Filters”, http://leadinganswers.typepad.com/leading_answers/files/agile_suitability_filters.pdf
- [Henrik Kniberg and Mattias Skarin](#), „Kanban and Scrum – making the most of both“, 2009, <http://www.infoq.com/minibooks/kanban-scrum-minibook>